Docket No.: 057454-0060 **PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of : Customer Number: 20277

Minematsu, ISAO : Confirmation Number: 3710

Application No.: 09/819,990 : Group Art Unit: 2183

Filed: March 29, 2001 : Examiner: Daniel H. Pan

For: MICROPROCESSOR EXECUTING DATA TRANSFER BETWEEN MEMORY AND REGISTER AND DATA TRANSFER BETWEEN REGISTERS IN RESPONSE TO SINGLE PUSH/POP INSTRUCTION

RESPONSE TO NOTIFICATION OF NON-COMPLIANT APPEAL BRIEF

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In response to Notification of Non-Compliant Appeal Brief mailed on June 26, 2006, attached is a Supplemental Appeal Brief corrected in accordance with the Examiner's request. This Supplemental Appeal Brief accompanies the Applicant's Request for Reinstatement of Appeal filed on November 18, 2005.

In the Notification of Non-Compliant Appeal Brief mailed on June 26, 2006, the Examiner requires the Applicant to show where "a single instruction code having a single operation code" recited in claim 1 is described in the specification by referring to page and line numbers.

Claim 1 recites a microprocessor including:

a program control unit controlling fetch of an instruction code;

an instruction decode unit decoding said fetched instruction code;

an address operation unit operating an address of a memory on the basis of the result of decoding by said instruction decode unit; and

a data operation unit operating data on the basis of the result of decoding by said instruction decode unit, wherein

said data operation unit executes data transfer between registers and data transfer between said registers and said memory in correspondence to single said instruction code having a single operation code fetched by said program control unit.

It is respectfully submitted that the limitation requiring the data operation unit to execute data transfer between registers and data transfer between said registers and said memory in correspondence to single said instruction code having a single operation code fetched by said program control unit is described on page 12, lines 3-19 of the specification and in FIGS. 9A, 9B and 10.

Further, it is noted that the Examiner raised no objections to the claim language during the prosecution of the present application. Accordingly, he admitted that the claim language is adequately supported by the specification.

Moreover, the Examiner requested the Applicant to indicate where the specification describes each feature of each dependent claim. It is respectfully submitted that this request is improper.

In accordance with 37 CFR 41.37 (c)(1)(v), a concise explanation of the subject matter defined in each independent claim should be presented. The rules require that for each dependent claim, every **means plus function and step plus function** must be identified referring to the specification. However, the dependent claims 2-10 do not have means plus function or step plus function limitations.

To the extent necessary, a petition for an extension of time under 37 C.F.R. 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account 500417 and please credit any excess fees to such deposit account.

Respectfully submitted,

MCDERMOTT WILL & EMERY LLP

Alexander V. Yampolsky
Registration No. 36,324

600 13th Street, N.W.
Washington, DC 20005-3096
Phone: 202.756.8000 AVY:apr
Facsimile: 202.756.8087
Date: July 26, 2006

Please recognize our Customer No. 20277 as our correspondence address.

Docket No.: 057454-0060 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of : Customer Number: 20277

Isao MINEMATSU : Confirmation Number: 3710

Application No.: 09/819,990 : Tech Center Art Unit: 2183

Filed: March 29, 2001 : Examiner: Daniel H. Pan

For: MICROPROCESSOR EXECUTING DATA TRANSFER BETWEEN MEMORY AND REGISTER AND DATA TRANSFER BETWEEN REGISTERS IN RESPONSE TO SINGLE PUSH/POP INSTRUCTION

SUPPLEMENTAL APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Supplemental Appeal Brief accompanies a Request for Reinstatement of Appeal.

Real Party In Interest

The real party in interest is Renesas Technology Corporation, the assignee of the entire right, title and interest in and to the above-identified U. S. Application.

Related Appeals and Interferences

No other appeals or interferences are known to the Appellant, which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

Status of Claims

Claims 1-10 are pending. Claims 11-14 are cancelled. Claims 1-10 stand under final rejection, from which rejection this appeal is taken.

Status of Amendments

According to the Advisory Action mailed April 11, 2005, the Amendment filed on March 21, 2005 will be entered for purposes of appeal.

Summary of Claimed Subject Matter

The claimed subject matter relates to a microprocessor capable of transferring data stored in multiple registers using a program having a reduced number of steps. As shown in FIG. 2 and described on pages 7-10, the microprocessor comprises an instruction decode unit 39 decoding an instruction fetched from an instruction memory 43, a program control unit (PCU) 40 controlling the fetch of instructions stored in the instruction memory 43, an address arithmetic unit (AAU) 41 producing addresses for accessing an X memory 44 or a Y memory 45 that store data, and a data arithmetic unit (DAU) 42 performing operations with data. In particular, the DAU 42 includes a multiplier 49, an arithmetic and logic unit (ALU) 50 and a shifter 51.

The microprocessor includes multiple registers illustrated in FIGS. 1A and 1B. In particular, the PCU 40 includes a register group 60 (FIG. 2) of 13 registers PC, PSW, BPC, BPSW, DPC, DPSW, PCLINK, LP_CT, REP_CT, LP_S, LP_E, PC_BRK and INT_S. The AAU 41 includes a register group 61 of 13 registers AR0 to AR3, AMD0 to AMD3, AR_SEL, MOD_S, MOD_E, SP and AR_PAGE. The DAU 42 includes a register group 62 of four registers TR0 to TR3, a register group 63 of four registers R0 to R3 and a register group 64 of two registers A0 and A1. As described on pages 5-7, different types of registers are provided. For example, four work registers TR0 to TR3 temporarily hold addresses or data, four address registers AR0 to AR3 store addresses for memory access, four operation source registers R0 to R3 store operation source data.

The microprocessor further includes data buses D1-D6 for transferring data between the registers. Data buses 53, 54 and 55 are provided for transferring data between the registers and the X memory 44 or the Y memory 45.

The DAU 42 performs operations based on control signal D (reference number 48 in FIG. 2). For example, the control signal D may instruct the DAU 42 to perform reading from the X memory 44 or Y memory 45, or perform operations with content of the registers and writing results into the X memory 44 or the Y memory 45.

As described on page 12, lines 3-19 of the specification and illustrated in FIGS. 9A, 9B and 10, the DAU 42 executes data transfer between the registers, and data transfer between the registers and the memory 44 or 45 in accordance with a single instruction having a single operation code fetched by the PCU 40. In particular, data from a control register to a work register and from the work register to the X memory 44 are transferred based on a single push instruction, and data from the X memory 44 to a work register and from the work register to a control register are transferred based on a single pop instruction.

In particular, FIGS. 9A and 9B illustrate exemplary POP and PUSH instructions. For example, when the POP instruction (2) in FIG. 9A specifies a register, the value of the work register TR0 is transferred to the register specified by the POP instruction, data stored at the address of the X memory 44 specified by the stack pointer is transferred to the work register TR0 through the data buses 53 and 54, and thereafter the value of the stack pointer is incremented. Further, when the PUSH instruction (2) in FIG. 9B specifies a register, data stored in the work register TR0 is loaded at the address of the X memory 44 specified by the stack pointer, the value of the register specified by the PUSH instruction is transferred to the work register TR0, and thereafter the value of the stack pointer is decremented.

FIG. 10 illustrates an exemplary program for pushing data stored in the operation source register R0 and the address register AR0 and thereafter popping the data. For example, "push R0" shown as instruction (2) in FIG. 10 indicates that the value of the work register TR0 is loaded at the address of the X memory 44 indicated by the stack pointer, the value of the operation source register R0 is transferred to the work register TR0 and thereafter the value of the stack pointer is decremented. "Push AR0" shown as instruction (3) in FIG. 10 indicates that the value of the work register TR0 is loaded at the address of the X memory 44 indicated by the stack pointer, the value of the address register AR0 is transferred to the work register TR0 and thereafter the value of the stack pointer is decremented. "Pop AR0" shown as instruction (6) in FIG. 10 indicates that the value of the work register TR0 is transferred to the address register AR0, data stored at the address of the X memory 44 indicated by the stack pointer is transferred to the work register TR0 and thereafter the value of the stack pointer is incremented. "Pop R0" shown as instruction (7) in FIG. 10 indicates that the value of the work register TR0 is transferred to the operation source register R0, data stored at the address of the X memory 44 indicated by the stack pointer is transferred to the work register TR0 and thereafter the value of the stack pointer is incremented.

Hence, in accordance with a claimed invention, a single instruction having a single operation code is sufficient to carry out data transfer between different registers as well as data transfer between the registers and the memory. As a result, the microprocessor is capable of performing operations with data held in multiple registers using a program having a reduced number of instructions.

Independent claim 1 is presented below with elements read on FIGS. 1, 2 and 4-11 of the drawings.

A microprocessor including:

a program control unit (40) controlling fetch of an instruction code;

an instruction decode unit (39) decoding said fetched instruction code;

an address operation unit (41) operating an address of a memory (44 or 45) on the basis of the result of decoding by said instruction decode unit (39); and

a data operation unit (42) operating data on the basis of the result of decoding by said instruction decode unit (39), wherein

said data operation unit (42) executes data transfer between registers (FIGS. 1A and 1B) and data transfer between said registers (FIGS. 1A and 1B) and said memory (44 or 45) in correspondence to single said instruction code having a single operation code fetched by said program control unit (FIGS. 9A, 9B and 10; page 12, lines 3-19 of the specification).

Grounds of Rejection To Be Reviewed By Appeal

1. Whether claim 1 is anticipated by Boggs et al. (5,687,338) under 35 U.S.C. 102(b).

2. Whether claims 2-10 are unpatentable over Boggs et al. (5,687,338) in view of Kudo et al. (6,560,692) under 35 U.S.C. § 103(a).

3. Whether claim 1 is unpatentable over Morrison (5,918,031) in view of Geldman et al. (5,524,268) under 35 U.S.C. § 103(a).

4. Whether claim 1 is unpatentable over Kudo et al. (6,560,692) in view of the MCS-8080/8085 Family User's Manual under 35 U.S.C. § 103(a).

Argument

1. Rejection of claim 1 as being anticipated by Boggs et al. (5,687,338) under 35 U.S.C. 102(b).

Anticipation, under 35 U.S.C. § 102, requires that each element of a claim in issue be found, either expressly described or under principles of inherency, in a single prior art reference. ***Kalman v. Kimberly-Clark Corp.,*** 713 F.2d 760, 218 USPQ 781 (Fed. Cir. 1983); ***Richardson v. Suzuki Motor Co.,*** 868 F.2d 1226, 9 USPQ2d 1920 (Fed. Cir. 1989) *cert. denied,* 110 S.Ct. 154 (1989). The term "anticipation," in the sense of 35 U.S.C. 102, has acquired the accepted definition meaning "the disclosure in the prior art of a thing substantially identical with the claimed invention." ***In re Schaumann***, 572 F.2d 312, 197 USPQ 5 (CCPA 1978). The initial burden of establishing a basis for denying patentability to a claimed invention rests upon the Examiner. ***In re Fine***, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988); ***In re Thorpe***, 777 F.2d 695, 227 USPQ 964 (Fed. Cir. 1985); ***In re Piasecki***, 745 F.2d 1468, 223 USPQ 785 (Fed. Cir. 1984). To satisfy this burden, therefore, each and every element of the claimed invention must be shown by the Examiner to be disclosed in Boggs et al. Appellant respectfully asserts that the record fails to meet this requirement.

In particular, claim 1 recites a microprocessor including:

- a program control unit controlling fetch of an instruction code;

- an instruction decode unit decoding said fetched instruction code;

- an address operation unit operating an address of a memory on the basis of the result of decoding by said instruction decode unit; and

- a data operation unit operating data on the basis of the result of decoding by said instruction decode unit.

The claim specifies that the data operation unit executes data transfer between registers and data transfer between the registers and the memory in correspondence to single said instruction code having a single operation code fetched by the program control unit.

Considering Boggs, the reference discloses macro instructions, which are broken into a number of micro instructions or micro operations during decoding (see col. 6, lines 20+).

The Examiner considers such a macro instruction to correspond to the claimed single instruction code for performing transfer between registers and transfer between the registers and the memory. In particular, the Examiner refers to the FAR_CALL instruction and the ADD instruction described in columns 6 and 7.

However, Boggs et al. does not disclose that the FAR-CALL or ADD instruction is a single instruction code having a single operation code, as claim 1 requires.

In the event the Examiner relied upon inherency without expressly indicating such reliance, the Examiner should be aware that inherency requires certainty, not speculation. ***In re Rijckaert***, 9 F.3rd 1531, 28 USPQ2d 1955 (Fed. Cir. 1993); ***In re King***, 801 F.2d 1324, 231 USPQ 136 (Fed. Cir. 1986); ***W. L. Gore & Associates, Inc. v. Garlock, Inc.***, 721 F.2d 1540, 220 USPQ 303 (Fed. Cir. 1983); ***In re Oelrich***, 666 F.2d 578, 212 USPQ 323 (CCPA 1981); ***In re Wilding***, 535 F.2d 631, 190 USPQ 59 (CCPA 1976). To establish inherency, the extrinsic evidence must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill. Inherency, however, may not be established by probability or possibilities. ***In re Robertson***, 169 F.3d 743, 49 USPQ2d 1949, 1950-51 (Fed. Cir. 1999).

The Examiner provided no factual basis upon which to conclude that the FAR-CALL or ADD macro instruction is a single instruction code having a single operation code, as claim 1 requires.

Moreover, one skilled in the would realize that a macro instruction of Boggs does not correspond to a single operation. Therefore, it cannot have a single operation code, as claim 1 requires.

In particular, the FAR_CALL instruction includes a micro code sequence composed of multiple micro operations listed in the table bridging column 6 and column 7. Further, the ADD instruction is decoded into a micro code sequence composed of multiple micro operations shown in col. 6, lines 32-37.

Hence, Boggs neither expressly nor inherently discloses the data operation unit that executes data transfer between registers and data transfer between the registers and the memory in correspondence to a single instruction code having a single operation code fetched by the program control unit, as claim 1 requires.

Therefore, the rejection of claim 1 as being anticipated by Boggs et al. under 35 U.S.C. 102(b) is improper.

2. Rejection of claims 2-10 over Boggs et al. (5,687,338) in view of Kudo et al. (6,560,692) under 35 U.S.C. § 103(a).

First, claims 2-10 depend from claim 1 and are defined over the reference combination at least for the reasons presented above in connection with claim 1.

Moreover, in the application of a rejection under 35 U.S.C. §103, it is incumbent upon the Examiner to factually support a conclusion of obviousness. As stated in ***Graham v. John Deere Co.*** 383 U.S. 1, 13, 148 U.S.P.Q. 459, 465 (1966), obviousness under 35 U.S.C. §103 must be determined by considering (1) the scope and content of the prior art; (2) ascertaining the differences between the prior art and the claims in issue; and (3) resolving the level of ordinary skill in the pertinent art.

However, the Examiner has failed to ascertain the differences between the prior art and the claims in issue.

In particular, the references do not teach or suggest that:

- the data operation unit transfers data stored in a first register to said memory and transfers data stored in a second register to said first register in correspondence to a single push instruction fetched by said program control unit, as claim 2 requires;

- the data operation unit decrements the value of a stack pointer after transferring the data stored in said second register to said first register, as claim 3 recites;

- the first register is a work register implemented in said data operation unit, as claims 4 and 8 recite;

- the second register is a control register implemented in one of said address operation unit and said program control unit, as claim 5 and 9 recite;

- the data operation unit transfers data stored in a first register to a second register and transfers data stored in said memory to said first register in correspondence to a single pop instruction fetched by said program control unit, as claim 6 recites;

- the data operation unit increments the value of a stack pointer after transferring said data stored in said memory to said first register, as claim 7 recites;

- the data operation unit transfers data stored in a first register to said memory and keeps the value of a stack pointer unchanged for a single push instruction fetched by said program control unit, as claim 10 recites.

It is noted that the Examiner's remarks in connection with the rejected dependent claims are unclear.

For example, in connection with claims 2, 8, he asserts that "first register of Kudo was a working register (general purpose)". In connection with claims 5, 9, the Examiner states that "Kudo's second register was a control register (special purpose)".

3. Rejection of claim 1 over Morrison (5,918,031) in view of Geldman et al. (5,524,268) under 35 U.S.C. § 103(a).

The Examiner admits that Morrison does not disclose the claimed single instruction having a single operation code.

Moreover, the Examiner admits that Morrison does not disclose data transfer between registers.

Geldman is relied upon for disclosing a single code for transfer between registers.

Considering Geldman, the reference discloses instruction MV8 for data transfer between registers (col. 8, lines 7-19). The Examiner considers this instruction to be a single instruction with a single operation code.

However, the reference specifically indicates that the instruction MV8 is composed of multiple operation codes (see table in col. 8, lines 10-15).

Further, in the application of a rejection under 35 U.S.C. §103, the Examiner must provide reasons why one having ordinary skill in the art would have been led to modify the prior art or to combine prior art references to arrive at the claimed invention. ***Ashland Oil, Inc. v. Delta Resins & Refractories, Inc.***, 776 F.2d 281, 227 USPQ 657 (Fed. Cir. 1985). ***In re Fine***, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988); ***Stratoflex, Inc. v. Aeroquip Corp.***, 713 F.2d 1530, 218 USPQ 871 (Fed. Cir. 1983); ***In re Warner***, 379 F.2d 1011, 154 USPQ 173 (CCPA 1967). These showings by the Examiner are an essential part of complying with the burden of presenting a *prima facie* case of obviousness. ***In re Oetiker***, 977 F.2d 1443, 1445, 24 USPQ2d 1443, 1444 (Fed. Cir. 1992).

The Examiner takes the position that "the use of Geldman could provide Morrison the ability (sic) to operate more than one operation transfer into a single instruction format, therefore, reducing the hardware space of the system." The Examiner does not explain the meaning of the term "hardware space of the system."

It is respectfully submitted that this position is unwarranted.

First, Morrison discloses collapsing two microcode sequences into a single micro operation. Therefore, one skilled in the art would realize that Morrison does not need "to operate more than one operation transfer into a single instruction format."

Further, one skilled in the art would realize that using a single instruction format for more than one operation transfer does not necessarily simplify the hardware of the computer system. Therefore, it is not apparent why one skilled in the art would have recognized any advantage to be gained by the proposed modification of Morrison in view of Geldman.

Moreover, Morrison does not need to transfer data between registers before loading the data to the memory (LOAD) or storing the data in the memory (STORE). Therefore, no reason is apparent to support the conclusion that one having ordinary skill in the art would have been impelled to add the data transfer between registers of Geldman to Morrison operations.

Further, even if Morrison were modified in view of the Geldman teaching, the claimed invention would not result.

In particular, neither Morrison nor Geldman suggests <u>two data transfer operations,</u> one of which is data transfer between registers and another - is data transfer between the registers and the memory, <u>in correspondence to a single instruction code</u> having a single operation code.

Therefore, if Morrison were modified in view of Geldman, the combined teachings of the Morrison and Geldman references would suggest performing a data transfer operation between registers in accordance with one instruction, and performing a data transfer operation between the registers and the memory in accordance with another instruction.

It is well settled that the test for obviousness is what the combined teachings of the references would have suggested to those having ordinary skill in the art. ***Cable Electric Products, Inc. v. Genmark, Inc.***, 770 F.2d 1015, 226 USPQ 881 (Fed. Cir. 1985). In determining whether a case of prima facie obviousness exists, it is necessary to ascertain whether the prior art teachings appear to be sufficient to one of ordinary skill in the art to suggest making the claimed substitution or other modification. ***In re Lalu***, 747 F.2d 703, 705, 223 USPQ 1257, 1258 (Fed. Cir. 1984).

As demonstrated above, the combined teachings of Morrison and Geldman are not sufficient to arrive at the claimed invention.

The Examiner has apparently failed to give adequate consideration to the particular problems and solution addressed by the claimed invention. ***Northern Telecom, Inc. v. Datapoint Corp.***, 908 F.2d 931, 15 USPQ2d 1321 (Fed. Cir. 1990); ***In re Rothermel***, 276 F.2d 393, 125 USPQ 328 (CCPA 1960). Specifically, as discussed above, the claimed invention suggests performing a data transfer between registers and a data transfer between the registers and a memory in accordance with a single instruction to simplify the program for controlling data transfer operations.

Neither Morrison nor Geldman addresses the problem and solution addressed by the claimed invention.

Accordingly, the Examiner's conclusion of obviousness is not warranted. Therefore, the rejection of claim 1 over Morrison in view of Geldman et al. under 35 U.S.C. § 103(a) is improper.

It is noted that in connection with this rejection, the Examiner takes the position that the applicant's specification does not disclose a single instruction code.

It is respectfully submitted that the Examiner's position is improper.

First, the Examiner never raised any objection or rejection relating to insufficient support of the claimed language by the specification.

Second, as discussed above in the Summary of the Claimed Subject Matter, the specification discloses a single instruction having a single operation code for data transfer between registers and data transfer between the registers and the memory.

4. Rejection of claim 1 over Kudo et al. (6,560,692) in view of the MCS-8080/8085 Family User's Manual under 35 U.S.C. § 103(a)

The Examiner admits that Kudo does not disclose data transfer between registers and between register and the memory in correspondence to a single instruction with a single operation code.

The User's Manual is relied upon for disclosing data transfer between register rh and memory (SP-1) and between register SP and r register in accordance with a single instruction code PUS rp.

Considering this reference, no PUS rp instruction is found. Possibly, the Examiner relied upon PUSH rp instruction disclosed on page 5-15.

This instruction is used to transfer data between registers of register pair rp and memory. No data transfer between registers is suggested.

It appears that the Examiner misunderstood the manual, which indicates that in response to the instruction PUSH rp the content of the high-order register of register pair pr is moved to the memory location whose address is one less than the content of register SP. The content of the low-order register of register pair pr is moved to the memory location whose address is two less than the content of register SP. The content of the register SP is decremented by 2 (see page 5-15).

Accordingly, the instruction PUSH rp does not provide data transfer between registers, as the Examiner asserts. Moreover, the reference does not disclose that the instruction PUSH rp has a single operation code.

Therefore, none of the applied references teaches or suggest data transfer between registers and data transfer between the registers and the memory in correspondence to a single instruction code having a single operation code, as claim 1 requires.

Hence, a combination of the applied references is not sufficient to arrive at the claimed invention. Accordingly, the rejection of claim 1 over Kudo et al. in view of the MCS-8080/8085 Family User's Manual under 35 U.S.C. § 103(a) is improper.

Conclusion

For all of the foregoing reason, Appellant respectfully submits that all rejections of the claims are improper and should be reversed.

Respectfully submitted,

McDERMOTT WILL & EMERY LLP

Alexander V. Yampolsky
Registration No. 36,324

600 13th Street, N.W.
Washington, DC 20005-3096
Phone: 202.756.8000 SAB/AVY/apr
Facsimile: 202.756.8087
Date: July 26, 2006

Please recognize our Customer No. 20277 as our correspondence address.

CLAIMS APPENDIX

1. (Original) A microprocessor including:

a program control unit controlling fetch of an instruction code;

an instruction decode unit decoding said fetched instruction code;

an address operation unit operating an address of a memory on the basis of the result of decoding by said instruction decode unit; and

a data operation unit operating data on the basis of the result of decoding by said instruction decode unit, wherein

said data operation unit executes data transfer between registers and data transfer between said registers and said memory in correspondence to single said instruction code having a single operation code fetched by said program control unit.

2. (Original) The microprocessor according to claim 1, wherein said data operation unit transfers data stored in a first register to said memory and transfers data stored in a second register to said first register in correspondence to a single push instruction fetched by said program control unit.

3. (Original) The microprocessor according to claim 2, wherein said data operation unit decrements the value of a stack pointer after transferring said data stored in said second register to said first register.

4. (Original) The microprocessor according to claim 2, wherein said first register is a work register implemented in said data operation unit.

5. (Original) The microprocessor according to claim 2, wherein said second register is a control register implemented in one of said address operation unit and said program control unit.

6. (Original) The microprocessor according to claim 1, wherein said data operation unit transfers data stored in a first register to a second register and transfers data stored in said memory to said first register in correspondence to a single pop instruction fetched by said program control unit.

7. (Original) The microprocessor according to claim 6, wherein said data operation unit increments the value of a stack pointer after transferring said data stored in said memory to said first register.

8. (Original) The microprocessor according to claim 6, wherein said first register is a work register implemented in said data operation unit.

9. (Original) The microprocessor according to claim 6, wherein said second register is a control register implemented in one of said address operation unit and said program control unit.

10. (Original) The microprocessor according to claim 1, wherein said data operation unit transfers data stored in a first register to said memory and keeps the value of a stack pointer unchanged for a single push instruction fetched by said program control unit.